

# CELLULAR AUTOMATA AND COMPUTER GRAPHICS

Şen ÇAKIR, Aylin KANTARCI

Ege University, Faculty of Engineering, Department of Computer Engineering, Bornova/İzmir

## ABSTRACT

Cellular Automata (CA) are simple mathematical systems which provide models for a variety of physical processes. They show how minute changes and simple rules lead to enormous changes in the behaviour of a system. They can also be used as computer graphics tools to produce a rich reservoir of interesting figures. In recent years, CA have attracted the attention of many scientists. Today, CA are used in many fields from ecology to image processing. In this paper, it is shown that a large number of complex and interesting patterns can be created with relatively simple CA rules.

**Key Words:** Cellular automata, Computer graphics

## HÜCRESEL OTOMATA VE BİLGİSAYAR GRAFİKLERİ

### ÖZET

Hücresele Otomata (CA) çeşitli fiziksel işlemler için model temin eden basit matematik sistemleridir. Önemsiz veya küçük değişikliklerin ve basit kuralların sistemlerin davranışlarında nasıl çok büyük değişikliklere yol açtığını gösterirler. İlginç şekillerin zengin bir kaynağı olarak bilgisayar grafikleri aracı olarak da kullanılmaktadırlar. Son yıllarda CA birçok bilim adamının dikkatini çekmiştir. Günümüzde CA ekolojiden görüntü işleme kadar birçok alanda kullanılır. Bu makalede oldukça basit CA kuralları ile çok sayıda karmaşık ve harika modelin yaratılabileceği gösterilmektedir.

**Anahtar Kelimeler :** Hücresele otomata, Bilgisayar grafikleri

### 1. INTRODUCTION

A dynamic system is a process which evolves with time. Some dynamic systems exhibit irregular behaviour. Minute changes in inputs of such systems may result in enormous differences in the outputs. The term 'chaos' is used to describe this kind of unstable behaviour. Cellular Automata (CA) are useful to demonstrate the chaotic behaviour of dynamic systems.

CA are great models to analyse many natural phenomena which exist in many fields. For example, the interaction among ants to form a colony or the interaction of water molecules to form a fluid can be modelled with CA. Most importantly, CA are suitable to define natural processes in computational terms, because, they consist of a finite grid of cells

with a finite number of states which evolve in discrete time steps, that is CA are discrete dynamical systems (Gutowitz, 1995).

CA have roots back in science. Pascal's triangle is considered to be the first CA. CA's recent development is related to the invention of computing machines. Konrad Zuse, Stanislaw Ulam, John Von Neumann and Stephan Wolfram are the first scientists to handle CA with today's view (Peitgen et al., 1992).

Today, CA have become a very important modelling and simulation tool in science and technology from physics, chemistry and biology, to computational fluid dynamics in airplane and ship design, philosophy and society. Because of their discreteness property, CA are suitable to design

parallel algorithms. Their parallel evolution provides performance improvements in simulations. For example, a traffic simulation takes only a few minutes with the use of CA models. On the other hand, simulations that use classical methods based on counting the number of cars may take many days (Wagner, 1995).

Mathematical properties of CA are popular research subjects among mathematicians. CA exhibit traits of self organisation, periodic stability and self replication. These features have applications in computer graphics. Simple rules which provide the evolution of CA result in complex and beautiful pictures (Peitgen and Saupe, 1989).

This paper is related to applications of CA in computer graphics. In Section 2, we surveyed the CA systems. Section 3 is on the applications of CA in computer graphics. The implementation is given in Section 4. Section 5 is the conclusion.

## 2. BACKGROUND

### 2. 1. Cellular Automata

A cellular automaton (CA) is an array of cells which interact with each other. A CA is defined by its state, a set of rules, its neighbourhood and an initial configuration. The state of a cell can be a number or a property. For instance, if each cell of a CA represents a part of a street, the state may represent the number of cars at that location. Rules define how cells interact with each other to change their states. Neighbourhood is a group of cells in interaction. Initial configuration of a CA is the initial state of its cells (Green, 1993).

CA are deterministic, local and dynamic systems. A CA is deterministic, because given the initial configuration and a rule, future states can be uniquely determined. A CA is local, The state of a cell at time t+1 is determined by a rule which is a function of its own state and the states of its neighbourhood at time t. A CA is dynamic because it evolves in discrete time steps. The rule is applied to all cells in synchrony. The state of cells are updated simultaneously and independently from one another.

A CA occupies a position in space and it is organised according to a specific geometry. A CA can be 1, 2 or 3 dimensional. 1 dimensional configurations are organised as an array of cells. It is possible to draw the succeeding steps of a 1 dimensional CA one below another and to obtain layered forms. 2 dimensional CA are arranged as a

grid of cells. 3 dimensional solid forms are not as common as 1 or 2 dimensional organisations. Other arrangements such as honeycomb forms are also sometimes used (Peitgen et al., 1992).

Many rules can be chosen from a large set of rules. For example, for a 2-state 1 dimensional CA with a neighbourhood template that includes a cell and its immediate left and right cells, there will be  $2^3 = 8$  neighbourhood states (000, 001,....., 111) and a choice of two states to map to for each of those, yielding a total of  $2^{2^3} = 2^8 = 256$  possible rules. Let the following rule be chosen from the rule set, as given in Table 1.

Table 1. Look up Table for CA in Figure 1

Neighbourhood	Leads to
000	0
001	1
010	1
011	0
100	1
101	1
110	0
111	0

Assuming the future evolutions are drawn one below the other, the CA in Figure 1 is obtained.

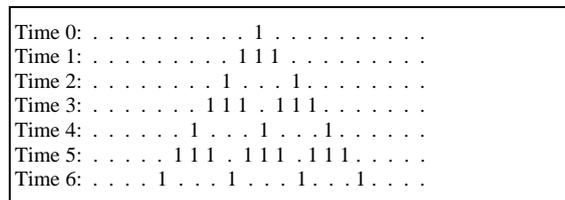


Figure 1. CA for the rule set given in Table 1. Here zero is denoted by “.”.

For a 2-state 2 dimensional template that includes a cell and its orthogonal neighbours, there will be  $2^5$  neighbourhood states and a choice of two states to map to for each of those, leading to  $2^{2^5} = 2^{32} = 4$  billion possible rules. Assume one of the possible rules has been chosen and presented in the following look up Table 2. The corresponding CA is given in Figure 2.

Table 2. Look up Table for the CA Given in Figure 2.

CSWNE	C	CSWNE	C	CSWNE	C	CSWNE	C
0000	0	01000	1	10000	1	11000	1
00001	0	01001	1	10001	1	11001	1
00010	0	01010	1	10010	1	11010	1
00011	0	01011	1	10011	1	11011	1
00100	1	01100	0	10100	1	11011	1
00101	1	01101	0	10101	1	11100	1
00110	1	01110	0	10110	1	11101	1
00111	1	01111	0	10111	1	11111	1

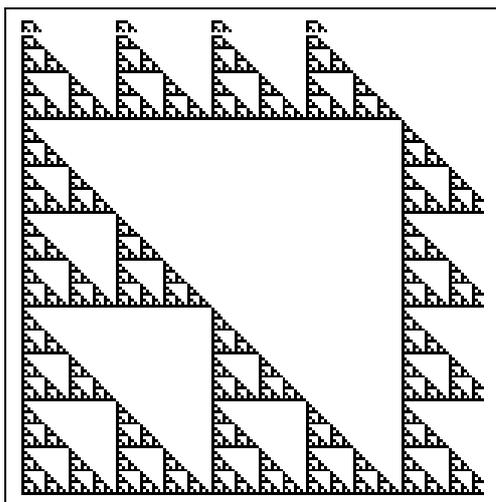


Figure 2. CA for the look up table given in Table 2.

## 2. 2. Properties of Cellular Automata

CA systems have the following properties (Green, 1993):

- 1) Self organisation: To simulate the behaviour of a cellular automaton; an initial configuration is chosen and CA rules are applied in discrete steps. If the resulting sequence of steps is observed, it is seen that iteration leads to an equilibrium behaviour which is independent of the choice of the initial configuration. Even if the system starts with a random arrangement of states, the rules force patterns to emerge.
- 2) Life like behaviour: Studies show that even the simplest linear CA behave in ways similar to complex biological systems. For example, the fate of any initial configuration of a CA is either
  - a. to die out,
  - b. to become stable or cycle with fixed periods,
  - c. to grow indefinitely at a fixed speed,
  - d. to grow and contract irregularly.
- 3) Thermal behaviour: In general, models that force a change of state for few configurations, tend to freeze into fixed patterns, whereas models that change the cells' states in most combinations tend to behave in a more active gaseous way.
- 4) Parallel evolution: At each time step, the cells change their states simultaneously; meaning that CA evolve in a parallel manner. This property provides performance improvements in simulations. CA are useful in designing parallel processing algorithms due to their parallel evolution property.

Properties of CA can easily be observed in a very popular CA called the Game of Life, invented in the 1970s by John Horton Conway. Game of Life is a 2 dimensional analogue of basic process in living systems. A cell in a 2 dimensional grid is either dead (0) or alive (1). The state of each cell changes from one generation to the next depending on the state of its immediate neighbours and its own state. The rules governing these changes are designed to mimic population change. *Life like behaviour* of the Game of Life can easily be seen by examining its evolution. A cell which is alive at one step will remain alive in the next step when precisely 2 or 3 cells among its 8 neighbours in a square lattice are alive. If more than 3 neighbours are alive, the cell will die from overcrowding. If fewer than 2 neighbours are alive, the cell will die from loneliness. A dead cell will come to life when surrounded by exactly 3 live neighbours. All cells change their states simultaneously, which means that Game of Life has a *parallel evolution*. Starting from an arbitrary initial pattern, the rules produce patterns quickly. This is the proof of the *self organisation* in Game of Life. Configurations either disappear entirely or break up into isolated patterns that are either static or cycle between different forms with a fixed period, being the indicator of the *thermal behaviour* of Game of Life (Green, 1993).

Some interesting patterns are observed throughout the evolution of Game of Life. Those are namely, blinkers, starships and guns. *Blinkers* are a group of cells which reproduces themselves periodically. *Gliders* move in certain directions. *Starships* leave a trace of blinkers. *Guns* periodically eject gliders (Peitgen et al., 1992).

## 3. CELLULAR AUTOMATA AND COMPUTER GRAPHICS

The increasing prominence of computer graphics has led to the rapid development of CA systems. Computer graphics is important for CA studies for two reasons: First CA are perfect tools for simulating natural phenomena and evolution of CA systems can easily be demonstrated with graphic tools. For example, reaction diffusion studies has progressed rapidly with CA simulations using computer graphics. The second reason is that CA represent appealing and mathematically interesting patterns. CA show how simple mathematical rules lead to a huge number of images which contain symmetry and stochasticity. Some CA are reminiscent of carpet designs, ceramic tile mosaics and brick patterns from Mosques.

CA figures contain mathematically interesting properties. Scaling symmetry is one of those properties. CA patterns are invariant under changes of scale. When examined carefully, it is noticed that the same basic shape is found at another place in another size. Scaling symmetry is also observed in many natural phenomena such as condensed matter physics, diffusion and polymer growth. That is why CA serve as visual and physical models for this type of structures. In addition to scaling symmetry, CA patterns contain infinite details, traits of self organization, periodic stability, etc.

With the growing popularity of CA in both computer graphics and the modeling of physical systems, an easy-to-use system for CA programming was needed. Dana Eckart from Radford University developed a CA programming language: Cellang. Cellang compiler generates code for both uniprocessor and shared memory multiprocessor systems. The entire system is written in C and the compiler produces C as an intermediate code, making the system highly portable. Cellang works under both UNIX and DOS operating system and currently supports the viewing of automata with either X Windows (UNIX), IRIS Graphics Library (UNIX),VGA(DOS) (Eckart, 1997).

#### 4. IMPLEMENTATION

In this section, some CA rules are introduced and the corresponding CA are given in Figure 4. The rules are implemented on a PC with C language.

The growth process can easily be observed and a different figure is obtained at each time step. The initial configuration is a random distribution of alive cells for the CA in Figure 4a. For other examples, the initial configuration is given in Figure 3. In the examples, 1s correspond to alive cells whereas 0s correspond to dead cells.

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Figure 3. Initial configuration for examples displayed in Figure 4b-4f

In Figure 4a, if five or more neighbours are alive, the cell becomes or remain alive. Otherwise, it will die or remain dead. Starting from a random distribution of alive cells, the system evolves as a stable pattern after some time steps. In the CA in Figure 4b, a cell becomes alive if only one of its

orthogonal neighbours is 1. One important characteristics of the system is that no cells die.

In Figure 4c, two rules are applied in order throughout the evolution of CA. Let n denote the number of time steps passed. The states of cells are determined at the (n + 1) th step according to the following rules.

1. A cell becomes alive if only one of its orthogonal neighbors is alive for even n.
2. A cell becomes alive if only one of its eight neighbors is alive for odd n .

The rule applied in Figure 4d is the same as the rule applied in Figure 4c except that the conditions that  $n \bmod 6 = 0$  and  $n \bmod 6 \neq 0$  determine the evolution of the pattern.

In the CA given in Figure 4e, a cell becomes alive if only one of its neighbours is alive. Otherwise, it is unchanged.

In Figure 4f, a cell and its orthogonal neighbors are taken into account. If the number of alive cells among these 5 cells is an even number, the cell dies or remain dead. Otherwise the cell becomes or remains alive. CAs in Figure 5 are obtained from the combinations of rules of Figures 4b, 4c, 4e, 4f. The difference between the two patterns is due to using the rules in different orders with different numbers of iterations.

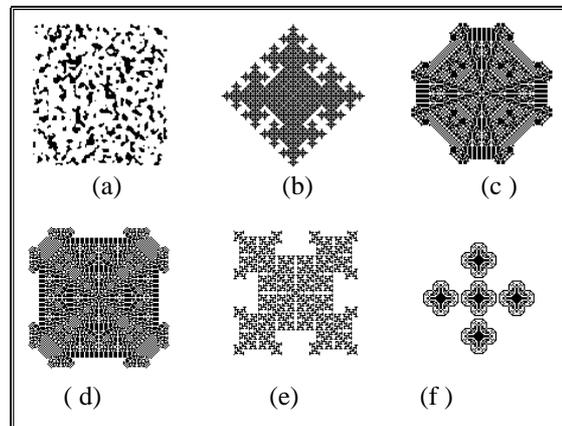


Figure 4. Examples of different CA rules

The initial state of CAn is random distribution of alive cells.

A CAn where no cells are allowed to die.

A CAn obtained by applying two rules in order.

Same CAn as given in (c) with the exception that the evolution of the pattern is determined by  $n \bmod 6$ .

A CAn where a cell becomes alive if only one of its neighbours is alive.

A CAn that takes different shapes at each time step.

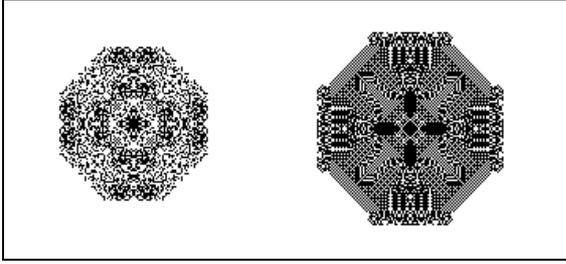


Figure 5. CA examples with different combinations of rules

## 5. CONCLUSIONS

CA provide a rich reservoir of beautiful and complex patterns. In contrast to the simplicity of rules, a large number of intricate figures are obtained. For example, for a system with two states and eight neighbors for a cell,  $2^{2^8} = 10^{154}$  rules can be defined.

CA portraits contain beauty and complexity which mathematicians were not able to fully appreciate before the age of computers (Pickover, 1989). Growth process can be easily observed by running CA on a high speed personal computer. CA can be run for a specified length of time. During the growth process, many different patterns for a single CA can be obtained at each step. In Figure 6, the growth of the CA given in Figure 4f is shown for time steps 1, 5, 7, 10, 15, 20, 24, 27, 30, 32, 37, 40, 43, 47, 50, 55, 58.

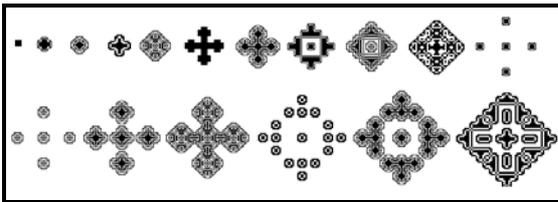


Figure 6. Growth of the CA given in Figure 4f

In Section 3, it is stated that CA patterns contain scaling symmetry and infinite details. Existence of scaling symmetry and infinite details is the main characteristic of fractals. Therefore, it can be concluded that CA are examples of fractals.

Another important feature of CA is their parallel evolution. At each time step, all cells change their states simultaneously according to their own states and the states of their neighbours at the previous step. Therefore, parallel programming environments speed up the evolution of CA patterns. Parallel CA programming is very promising for time consuming simulations of natural phenomena. Our study is continuing along this direction.

## 6. REFERENCES

- Eckart, D. 1997. Cellular Automata Home Page, <http://www.runet.edu/dana/ca/ca.html>, 15.5.1998, 15:00.
- Green, D. 1993. Cellular Automata Home Page, <http://life.csu.edu.au/complex/tutorials/tutorial1.html>, 15.5.1998, 15:00.
- Gutowitz, H. 1995. Cellular Automata and the Science of Complexity, Journal of Complexity.
- Peitgen, H., Saupe, D. 1989. *The Science Fractal Images*, 314 pages. Springer-Verlag Press, New York.
- Peitgen, H., Saupe, D., Jurgens H. 1992. *Chaos and Fractals: New Frontiers of Science*, 500 pages, Springer-Verlag Press, New York.
- Pickover, C. 1989. Mathematics and Beauty VIII, Tesselation Automata Derived from a Single Defect, Computer Math. Applic., (17), 321-336.
- Wagner, P. 1995. ZPR Home Page, The Traffic Simulation Group at the ZPR, [http://www.zpr.uni.koeln.de/GroupBachem/VERK\\_EHR.PG](http://www.zpr.uni.koeln.de/GroupBachem/VERK_EHR.PG), 15.5.1998, 15:00.